CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Elsevier - Publisher Connector

ELSEVIER

Theoretical Computer Science 170 (1996) 129–144

Theoretical Computer Science

# Tractability of cut-free Gentzen type propositional calculus with permutation inference

Noriko H. Arai *

*Department of Computer Science, Hiroshima City University, 151 Ozuka, Asaminami-ku, Hiroshima 731-31 Japan*

Received March 1995; revised November 1995
Communicated by M. Nivat

**Abstract**

We present a new propositional calculus that has desirable natures with respect to both automatic reasoning and computational complexity: we introduce an inference rule, called permutation, into a cut-free Gentzen type propositional calculus. It allows us to obtain a system which (1) guarantees the subformula property and (2) has polynomial size proofs for hard combinatorial problems, such as pigeonhole principles. We also discuss the relative efficiency of our system. Frege systems polynomially prove the partial consistency of our system.

## 1. Introduction

One of the most fundamental problems of the complexity theory and the automated reasoning theory is to find an efficient proof system for propositional calculus which is applicable for automated reasoning. The statement contains two intuitive concepts. First, we have to make it clear what the notion "efficient" means. There is a wide spread understanding that polynomial time computability is the correct mathematical model of feasible computation. According to the opinion, a truly "effective" system must have a polynomial size, $p(n)$ proof for every tautology of size $n$. In [9], Cook and Reckhow named such a system, a *super system*. They showed that if there exists a super system, then NP = co-NP; many people are highly skeptical about the validity of this equality. Secondly, we have to have some criteria for propositional calculi to be applicable for automatic theorem proving. Intuitively, we say that tautologies are automatically proved when we can construct a deterministic machine which says yes if the input is a tautology and says no otherwise. If we interpret our goal most strictly, we have to obtain a sound proof system which proves any tautology polynomially and

* E-mail: arai@cs.hiroshima-cu.ac.jp.

0304-3975/96/$15.00 © 1996 — Elsevier Science B.V. All rights reserved
*PII* S0304-3975(95)00241-3

the construction of the proof is completely determined by the structure of the tautology. Then, obviously, P = NP is necessary.

How can we relax our criteria so that it is theoretically meaningful but still practical? One fairly natural approach is to give up to prove every tautology polynomially but confine ourselves to "familiar" tautologies.

Gentzen's Hauptsatz suggests us that cut-free Gentzen type sequent calculus is one of the most reasonable systems to be applied to automatic reasoning: we can obtain a proof-tree automatically for any given tautology. Furthermore, the construction procedure can be determined solely by the structure of the given tautology. However, it is already known that the number of steps required in the search procedure increases exponentially with the length of inputs [14]. Resolution is another propositional calculus which is frequently mentioned in automatic theorem proving. It is also known that there are sequences of tautologies which require exponential size proofs [12]. Unfortunately, the hard examples of cut-free Gentzen system or for resolution are not rare nor pathological, but they are rather commonly found combinatorial problems [15].

We suggest another possible approach; if it is too much to ask to construct a deterministic machine accepting tautologies in polynomial time, it is worth trying to construct a nondeterministic machine but the chance to obtain a sound proof for a given tautology is relatively high. Gentzen system with cut-rule and Frege system are known to be a strictly more powerful system than resolution [3, 12]. However, they do not satisfy the subformula property: the existence of cut-rule and modus-ponens allows unpredictable formulas to coming into proofs. As a result, chance to obtain appropriate proofs by machine is very low even for simple tautologies. On the contrary, if a system satisfies the subformula property, the bound for search will be relatively limited.

It is sensible to note that many hard examples for propositional calculus such as pigeonhole principles are originally first-order sentences. Translating them into propositional formulas, these propositions share an evident similarity, symmetries. If we can express as an inference rule that a tautology remains invariant under permutation of variables, proofs of propositions of this kind can be shortened dramatically [2].

In this paper, we introduce a new inference rule to play the role: *permutation rule*. We first show that a cut-free Gentzen type sequent calculus plus permutation, called GCNF′ + permutation, satisfies the subformula property. Then, we show that the system has polynomial size proofs for both the pigeonhole principle and the $k$-equipartition.

## 2. Gentzen system GCNF′

**Definition 1.** *Resolution* proves a formula to be a tautology by showing that its negation, which is put into conjunctive normal form, is unsatisfiable.

A *propositional variable* is denoted by $p, q, r, x$. Each propositional variable has a conjugate (or negation) denoted by $\bar{p}$. Also $\bar{\bar{p}} = p$. A *literal* is a propositional variable $p$ or a conjugate $\bar{p}$. A *clause* is a finite set of literals, where the meaning

of the clause is the disjunction of the literals in the clause. For example $\{p_1, \bar{p}_2, p_3\}$ means $p_1 \vee \bar{p}_2 \vee p_3$.

Resolution has no axiom. It has only one inference rule called *resolution rule*:

$$\text{Resolution rule:} \quad \frac{C_1 \cup \{x\} \quad C_2 \cup \{\bar{x}\}}{C_1 \cup C_2}$$

When we try to show that a set of clauses **C** is unsatisfiable, we take **C** to be a set of hypotheses to which we apply the resolution rule until we obtain the empty clause.

*GCNF′* is a variant of cut-free Gentzen system introduced by Gallier (see [11, p. 120]). It is also a refuting system.

A *cedent* is a finite set of clauses, expressed as a sequence of clauses punctuated by commas. The meaning of a cedent is the conjunction of the clauses in the cedent. For example $C_1, C_2, \ldots C_n$ means $C_1 \wedge C_2 \wedge \ldots \wedge C_n$. We use capital Greek letters $\Gamma, \Delta, \Pi$ for cedents. The semantics of cedents implies that a cedent $C_1, \ldots, C_n$ is false iff the formula $C_1 \wedge \ldots \wedge C_n \supset \perp$ is valid.

Axioms: $p, \bar{p}$

$$\text{Structural inference:} \quad \frac{\Gamma}{\Gamma, \Delta}$$

$$\text{Logical inference:} \quad \frac{\Gamma, C_1, \ldots, C_k \quad \Pi, l}{\Gamma \cup \Pi, C_1 l, \ldots, C_k l}(l)$$

$l$ is an arbitrary literal, which is called the *auxiliary literal* of this inference.

It is fairly easy to show the soundness and the completeness of GCNF′ (see [11, Chap. 4]).

**Proposition 1.** GCNF′ *is sound and complete.*

Now we define a scale to measure the efficiency of a proof system.

**Definition 2.** 1. Let $S$ be a proof system which is sound and complete, and let $P$ be a proof system of $S$. The *size* of $P$ is the number of all the symbols used in $P$, denoted by $size(P)$.

2. Let $S_1$ and $S_2$ be proof systems for propositional calculus. $S_1$ *p-simulates* $S_2$ iff there exists a polynomial function $p$ such that for any formula $f$ and any proof $P_2$ of $f$ in $S_2$, there exists a $S_1$-proof $P_1$ of $f$ (translated into $S_1$ language) so that

$$size(P_1) \leqslant p(size(P_2))$$

A system $S_1$ $p$-simulates $S_2$ iff $S_1$ is not less efficient than $S_2$ as a proof system. GCNF′ in tree form and resolution in tree form polynomially simulate each other.

**Proposition 2.** 1. *Let $P$ be a tree* GCNF′ *refutation of $C_1, \ldots, C_n$. Then, there exists a tree resolution refutation $R$ of $C_1, \ldots, C_n$ with*

$$size(R) \leqslant size(P)$$

2. *Let $R$ be a tree resolution refutation of $C_1,\ldots,C_n$. Then, there exists a tree* GCNF′ *refutation $P$ of $C_1,\ldots,C_n$ with*

$$size(P) \leqslant size(P)^2$$

**Proof.** (1) We prove by the number of lines in $P$. Suppose that the last inference of $C_1,\ldots,C_n$ is a logical inference of the form,

$$\frac{\Gamma, C_1,\ldots,C_n \quad \Delta, l}{\Gamma \cup \Delta, C_1 l,\ldots,C_n l.}$$

Denote the subtrees up to $\Gamma, C_1,\ldots,C_n$ and $\Delta, l$ by $P_1$ and $P_2$. By the induction hypothesis, there are tree resolution refutation $R_1$ of $\Gamma, C_1,\ldots,C_n$ and $R_2$ of $\Delta, l$ with $size(R_1) \leqslant size(P_1)$ and $size(R_2) \leqslant size(P_2)$. On one hand, replace each leaf labeled by $C_1,\ldots,C_n$ in $R_1$ by $C_1 l,\ldots,C_n l$ to obtain a new tree $R_1'$. The root of $R_1'$ must be $l$ instead of an empty clause. On the other hand, delete every leaf labeled by $L$ in $R_2$ to obtain a new tree $R_2'$. The root of $R_2'$ is $\bar{l}$. Resolute the roots of $R_1'$ and $R_2'$ to have an empty clause. This new tree satisfies the condition. (2) is proved similarly.

It remains open whether cut-free Gentzen in a directed acyclic graph (DAG) $p$-simulates resolution in DAG form. Likewise, it is not known whether GCNF′ in DAG form $p$-simulates resolution in DAG form or not. It is subtle to answer the question if resolution in DAG form $p$-simulates cut-free Gentzen in DAG form. The answer depends on how we translate a formula including $\neg, \vee, \wedge$ and $\supset$ into conjunctive normal form. A traditional translation usually produces exponential size formulas comparing with the original formulas. Tseitin gave an answer in [13]: for a given formula $\varphi$, he introduced a set of new variables and assigned the variables to each subformula of $\varphi$. This method is called limited extension. Then, he gave a set $\mathbf{C}$ of clauses made of these new variables so that $\mathbf{C}$ is contradictory iff $\varphi$ is valid. He showed the following.

**Theorem 1** (Tseitin [13]). *Resolution in DAG form p-simulates cut-free Gentzen in DAG.*

If we use limited extension, it is obvious that resolution in DAG $p$-simulates GCNF′ in DAG. However, formulas of GCNF′ are already in conjunctive normal form; there is no need to translate. It remains unsolved if resolution in DAG form $p$-simulates GCNF′ in DAG form without using limited extension.

In the following argument, we understand proofs of GCNF′ or resolution to be in DAG form. If $P$ is a GCNF′ (resolution) proof, then $size(P)$ means the number of symbols appearing in different cedents (clauses) in $P$. Now we examine hard examples for GCNF′. Haken [12] showed an exponential lower bound for resolution: he proved that there exists a constant $c, c > 1$ so that, for sufficiently large $n$, every resolution refutation of the pigeonhole principle ($PHP_n$) contains at least $c^n$ different clauses. Ajtai [1] showed a superpolynomial lower bound for constant depth Frege proofs for the pigeonhole principle, and later showed a superpolynomial lower bound for constant

depth Frege proofs for 2-*Equipartition* even assuming the pigeonhole principle. Their proofs can be translated to prove a superpolynomial lower bound for GCNF′.

**Definition 3** (*Pigeonhole principle*). *The pigeonhole principle* states that for each $n$, if $f : \{0,\ldots,n\} \to \{0,\ldots,n-1\}$ then $f$ is not one-to-one.

For each $i$ and $j$ with $0 \leqslant i \leqslant n$ and $0 \leqslant j \leqslant n-1$ we will have the variable $p_{i,j}$ which 'means' $f(i) = j$.

$$PHP_n \qquad \bigwedge_{0\leqslant i\leqslant n} \bigvee_{0\leqslant j\leqslant n-1} p_{i,j}, \qquad \bigwedge_{0\leqslant i<m\leqslant n} \bigvee_{0\leqslant j\leqslant n-1} (\bar{p}_{i,j} \bar{p}_{m,j})$$

$\bigvee_{0\leqslant i\leqslant n} p_i$ is an abbreviation for the clause $p_0,\ldots,p_n$. $\bigwedge_{0\leqslant i\leqslant n} C_i$ is an abbreviation for the cedent $C_0,\ldots,C_n$.

The number of all literals contained in $PHP_n$ is $n^3 + 2n^2 + n$.

**Definition 4** (*k-equipartition*). *The k-equipartition* states that if an integer $n$ is not evenly divisible by $k$, then there is no partition of $\{1,\ldots,n\}$ into disjoint sets of size $k$.

Let $J_n^k = \{(j_1,\ldots,j_k) : 1 \leqslant j_1 < \ldots \vec{j}_k \leqslant n\}$ For $\vec{j} \in J$, we write $i \in \vec{j}$ to mean that there exists $1 \leqslant l \leqslant k$ such that $i = j_l$. Suppose that $n \not\equiv 0 \,(\mathrm{mod}\, k)$. We introduce new variables $x_{i,(j_1,\ldots,j_k)}$ for $1 \leqslant i, j_1,\ldots,j_k \leqslant n$ to mean that $(j_1,\ldots,j_k)$ is a partition of $\{1,\ldots,n\}$ and $i \in \{j_1,\ldots,j_k\}$.

$k$–$Eq(n)$ is defined as the following cedent;

$$\bigwedge_{1\leqslant i\leqslant n} \bigvee_{\vec{j}\in J_n^k, i\in\vec{j}} x_{i,\vec{j}}, \qquad \bigwedge_{\vec{j}\in J_n^k, i_1,i_2\in\vec{j}, i_1\neq i_2} (\bar{x}_{i_1,\vec{j}} x_{i_2,\vec{j}}), \qquad \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J_n^k \\ i\in\vec{j}_1, i\in\vec{j}_2, \vec{j}_1\neq\vec{j}_2}} (\bar{x}_{i,\vec{j}_1} \bar{x}_{i,\vec{j}_2})$$

The number of all literals contained in $k$–$Eq(n)$ is

$$n\binom{n-1}{k-1} + 2\binom{n}{k}\binom{k}{2} + n\binom{n-1}{k-1}^2 - n\binom{n-1}{k-1}.$$

The first $\bigwedge$ of clauses expresses that "each $i$ is contained in some partition whose size is $k$." The second $\bigwedge$ of clauses expresses that "if $(i_1,\ldots,i_k)$ is a partition containing $i_1$, then it is also a partition containing $i_2,\ldots$ and $i_k$." The last $\bigwedge$ of clauses means that "if $i_s = j_t$ for some $1 \leqslant s \leqslant k$ and $1 \leqslant t \leqslant k$ and if $(i_1,\ldots,i_k) \neq (j_1,\ldots,j_k)$, then either $(i_1,\ldots,i_k)$ or $(j_1,\ldots,j_k)$ is not a partition."

(Note: The definition given above is slightly different from the formulation given in [8], but they are equivalent.)

**Proposition 3** (Haken [12]). *There exists a constant $c$, $c > 1$ such that, for sufficiently large $n$, every* GCNF′ *refutation of $PHP_n$ contains at least $c^n$ different cedents.*

**Proposition 4** (Ajtai [1]). *There exists a constant $c$, $c > 1$ so that, for sufficiently large $n$, every GCNF′ refutation of $k$–Eq($n$) contains at least $c^n$ different cedents.*

We introduce new inference rules, called *renaming*, *restricted renaming* and *permutation.*

Renaming: $$\frac{\Gamma}{\Gamma(p \to q)}\, p \to q$$

$\Gamma(p \to q)$ is obtained by replacing every occurrence of $p$ by $q$ in $\Gamma$.

Restricted renaming: $$\frac{\Gamma}{\Gamma(p \Rightarrow q)}\, p \Rightarrow q$$

$\Gamma(p \Rightarrow q)$ is obtained by replacing every occurrence of $p$ in $\Gamma$ by a variable $q$, which does not appear in $\Gamma$.

Permutation: $$\frac{\Gamma(p_1, \ldots, p_m)}{\Gamma(\pi(p_1), \ldots, \pi(p_m))}\, \pi$$

$\pi$ is a permutation on $\{p_1, \ldots, p_m\}$ and $\Gamma(\pi(p_1), \ldots, \pi(p_m))$ is the result of replacing every occurrence of $p_i$, $1 \leqslant i \leqslant m$ in $\Gamma(p_1, \ldots, p_m)$ by $\pi(p_i)$.

It is straightforward to show that GCNF′ + restricted renaming $p$-simulates GCNF′ + permutation.

**Proposition 5.** GCNF′+ *restricted renaming p-simulate* GCNF′+*permutation.*

**Proof.** A permutation is a product of disjoint cycles. Hence, it is enough to consider the case that a given permutation is a cycle.

$$\frac{\Gamma(p_1, \ldots, p_{m-1}, p_m)}{\Gamma(p_2, \ldots, p_m, p_1)}$$

can be expressed as a sequence of $m + 1$ restricted renaming inferences;

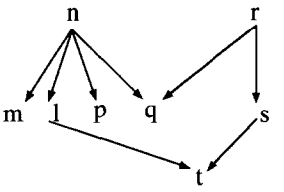
$$\frac{\Gamma}{\Gamma(p_m \Rightarrow x)}, \frac{\Gamma}{\Gamma(p_{m-1} \Rightarrow p_m)}, \ldots, \frac{\Gamma}{\Gamma(p_1 \Rightarrow p_2)}, \frac{\Gamma}{\Gamma(x \Rightarrow p_1)}$$

where $x$ is a variable not occurring in $\Gamma$.

In general, GCNF′+ permutation does not satisfy the subformula property. However, one can translate a given GCNF′+permutation refutation into a GCNF′ + permutation refutation satisfying the subformula property without increasing its size too much. Before we start, we need some definitions.

**Definition 5.** Let $D$ be a directed acyclic graph. Suppose that $n, m$ are nodes appearing in $D$. When $m$ appears below $n$ and no other node appears between $n$ and $m$, we say that $m$ is a *son* of $n$. When $n_1, \ldots, n_k$ are the sons of $n$, and when $n_1$ is the leftmost occurrence among them, we say that $n_1$ is the *direct son* of $n$. $n_2, \ldots, n_k$ are called *nondirect sons* of $n$. A sequence of nodes $m_1, \ldots, m_l$ is called a *direct line* of $n_l$ in $D$

when $n_1$ is either a leaf or a nondirect son of a node in $D$, and every $n_i$ for $1 < i \leqslant l$ is the direct son of $n_{i-1}$.




In the following, we frame a two-dimensional image of directed acyclic graphs so that we can fix the order of right and left of nodes.

**Theorem 2** (Subformula property of GCNF′+permutation). *Let $P$ be a* GCNF′+ *permutation refutation of $C_1,\ldots,C_n$. Then, there exists $P'$, a refutation of $C_1,\ldots,C_n$ such that size$(P')$ = O(size$(P)^3$) and $P'$ satisfies the subformula property; every clause $C = l_1,\ldots,l_m$ appearing in $P'$ is a subformula of one of $C_1,\ldots,C_n$.*

**Proof.** We shall transform $P$ into $P'$ inductively from the bottom to the top.

Suppose that $n$ is a node in $P$. Let $n_1,\ldots,n_l$ are the list of sons of $n$. Suppose that $n_1$ is the direct son of $n$. When

$$\frac{n}{n_1}$$

is weakening, no change is made. If

$$\frac{n \quad m}{n_1}$$

is a logical inference, no change is made. Suppose that the inference between $n$ and $n_1$ is permutation, say

$$\frac{\Gamma(p_1,\ldots,p_m)}{\Gamma(\pi(p_1),\ldots,\pi(p_m))}\pi$$

Then, replace every occurrence of $p_i$ by $\pi(p_i)$ $(1 \leqslant i \leqslant m)$ in each cedent on every direct line containing the upper cedent, $\Gamma(p_1,\ldots,p_m)$. The result may fail to be a GCNF′ + permutation refutation: there may exist a gap between a node and its nondirect son. Suppose that $n$ in $P$ is replaced by $n'$, and its nondirect son $n_k$ is replaced by $n'_k$. Suppose that the inference between $n$ and $n_k$ is a permutation. Note that a product of permutations is again a permutation. Hence,

$$\frac{n'}{n'_k}$$

is a sound permutation inference. Suppose that the inference between $n$ and $n_k$ is either structural or logical, then insert one permutation inference necessary. Now we obtain a sound GCNF′ + permutation refutation, $P'$.

We show that $P'$ satisfies the subformula property by induction on the construction of $P'$. Let $m$ be a node in $P'$. Let $m_1$ be the direct son of $m$. Then, by the induction hypothesis, $m_1$ satisfies the subformula property. The inference between $m$ and $m_1$ is either a logical inference, structural inference, or a special kind of restricted renaming, which is

$$\frac{\Gamma}{\Gamma.}$$

Hence, $m$ also satisfies the subformula property.

We remark that a close examination of the proof of Theorem 2 gives us a polynomial algorithm to translate a GCNF′ + permutation refutation to GCNF′ + permutation which satisfies the subformula property.

There is little hope for GCNF′ + renaming to enjoy the subformula property. We discuss later how powerful renaming inference is. It is not known if GCNF′ + restricted renaming enjoys the subformula property or not.

A resolution refutation $R$ is called *regular* iff for every resolution

$$\frac{C_1 \cup \{x\} \quad C_2 \cup \{\bar{x}\}}{C_1 \cup C_2}(I)$$

appearing in $R$, no resolution of the form,

$$\frac{D_1 \cup \{x\} \quad D_2 \cup \{\bar{x}\}}{D_1 \cup D_2}$$

appears below $I$. This notion was introduced by Tseitin [13]. He proved that regular resolution is not super before Haken's work. By analogy, we say a GCNF′ (or GCNF′ + permutation) refutation $P$ is *regular* iff for every logical inference $I$ whose auxiliary literal is $l$ in $P$, no logical inference having the same auxiliary literal $l$ appears below $I$.

We show that regular GCNF′+permutation has polynomial size refutations for $PHP_n$ and $k$–$Eq(n)$.

**Theorem 3.** *There exists a regular* GCNF′ + *permutation refutation of* $PHP_n$ *whose size* $\leqslant O(n^6)$.

**Proof.** Assume that we already have a regular GCNF′ + permutation refutation $P_{n-1}$ of

$$\bigvee_{j=0}^{n-2} p_{0,j}, \ldots, \bigvee_{j=0}^{n-2} p_{n-1,j}, \bigwedge_{0 \leqslant i < m \leqslant n-1} \bigwedge_{0 \leqslant j \leqslant n-2} (\bar{p}_{i,j} \bar{p}_{m,j})$$

such that $size(P_{n-1}) \leqslant O((n-1)^6)$. We supplement some lines below $P_{n-1}$ to obtain $P_{n,n-1}$. First, we add a logical inference of which auxiliary literal is $p_{n-1,n-1}$.

$$\frac{\bigvee_{j=0}^{n-2} p_{0,j},\ldots,\bigvee_{j=0}^{n-2} p_{n-1,j},\ \bigwedge_{0\leqslant i<m\leqslant n-1}\bigwedge_{0\leqslant j\leqslant n-2}(\bar{p}_{i,j}\bar{p}_{m,j})\quad \bar{p}_{n-1,n-1}, p_{n-1,n-1}}{\bar{p}_{n-1,n-1},\bigvee_{j=0}^{n-2} p_{0,j},\ldots,\bigvee_{j=0}^{n-2} p_{n-2,j}\bigvee_{j=0}^{n-1} p_{n-1,j},\ \bigwedge_{0\leqslant i<m\leqslant n-1}\bigwedge_{0\leqslant j\leqslant n-2}(\bar{p}_{i,j}\bar{p}_{m,j})}(p_{n-1,n-1})$$

Similarly, add logical inferences whose auxiliary literals are $p_{n-2,n-1},\ldots,p_{0,n-1}$, and whose right upper cedents are axioms. Then, we get

$$\bar{p}_{0,n-1},\ldots,\bar{p}_{n-1,n-1},\bigvee_{j=0}^{n-1} p_{0,j},\ldots,\bigvee_{j=0}^{n-1} p_{n-1,j},\ \bigwedge_{0\leqslant i<m\leqslant n-1}\bigwedge_{0\leqslant j\leqslant n-2}(\bar{p}_{i,j}\bar{p}_{m,j})$$

This refutation graph is called $P_{n,n-1}$. The last cedent means that "for all $0\leqslant k\leqslant n-1$, the $k$th pigeon sits in one of the holes. $0,\ldots,n-1$. At the same time, the pigeon does not sit in the $(n-1)$th hole." Define a permutation $\pi_k$ by a product of $(n-1)$ transpositions,

$$(p_{0,n-1}\quad p_{0,k})\cdots(p_{n-1,n-1}\quad p_{n-1,k})$$

for all $0\leqslant k\leqslant n-2$. To obtain $P_{n,k}$, for each $0\leqslant k\leqslant n-2$ add one permutation inference;

$$\frac{\bar{p}_{0,n-1},\ldots,\bar{p}_{n-1,n-1},\bigvee_{j=0}^{n-1} p_{0,j},\ldots,\bigvee_{j=0}^{n-1} p_{n-1,j},\ \bigwedge_{0\leqslant i<m\leqslant n-1}\bigwedge_{0\leqslant j\leqslant n-2}(\bar{p}_{i,j}\bar{p}_{m,j})}{\bar{p}_{0,k},\ldots,\bar{p}_{n-1,k},\bigvee_{j=0}^{n-1} p_{0,j},\ldots,\bigvee_{j=0}^{n-1} p_{n-1,j},\ \bigwedge_{0\leqslant i<m\leqslant n-1}\bigwedge_{\substack{0\leqslant j\leqslant n-1,\\ j\neq k}}(\bar{p}_{i,j}\bar{p}_{m,j})}\pi_k$$

For each $0\leqslant k\leqslant n-1$, we add a logical inference;

$$\frac{\bar{p}_{0,k},\ldots,\bar{p}_{n-1,k},\bigvee_{j=0}^{n-1} p_{0,j},\ldots,\bigvee_{j=0}^{n-1} p_{n-1,j},\ \bigwedge_{0\leqslant i<m\leqslant n-1}\bigwedge_{0\leqslant j\leqslant n-1,j\neq k}\bar{p}_{i,j}\bar{p}_{m,j}\quad p_{n,k},\bar{p}_{n,k}}{p_{n,k}\bigvee_{j=0}^{n-1} p_{0,j},\ldots,\bigvee_{j=0}^{n-1} p_{n-1,j},\ \bigwedge_{0\leqslant i<m\leqslant n-1}\bigwedge_{\substack{0\leqslant j\leqslant n-1\\ j\neq k}}\bar{p}_{i,j}\bar{p}_{m,j}\bigwedge_{0\leqslant i\leqslant n-1}\bar{p}_{n,k}\bar{p}_{i,k}}(\bar{p}_{n,k})$$

Combine these together by applying $n-1$ logical inferences to obtain $P_n$ of $PHP_n$. $P_{n-1}$ is regular by the induction hypothesis, so is $P_n$.

$$size(P_n)\leqslant(len(P_{n-1})+2n+2(n-1))(n+1+n^2(n+1)/2)\leqslant o(n^6).$$

**Theorem 4.** *There exists a polynomial function $p$, independent from $n$, and a regular* GCNF′+ *permutation refutation of $k$–$Eq(n)$ whose size is $\leqslant p(n)$.*

**Proof.** We prove by induction on $n$. If $k+1\leqslant n<2k$, it is obvious. Suppose that we already obtain a GCNF′+ permutation refutation $P_n$ for $k$–$Eq(n)$ such that $size(P_n)\leqslant p(n)$. $k$–$Eq(n+k)$ is a cedent expressed as a conjunction of the following:

1. $\bigwedge_{1\leqslant i\leqslant n+k}\bigvee_{\vec{j}\in J^k_{n+k},i\in\vec{j}} x_{i,\vec{j}}$

2. $\bigwedge_{\vec{j}\in J^k_{n+k}, i_1,i_2\in\vec{j}, i_1\neq i_2} (\bar{x}_{i_1,\vec{j}} x_{i_2,\vec{j}})$

3. $\bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_{n+k}\\ i\in\vec{j}_1, i\in\vec{j}_2, \vec{j}_1\neq\vec{j}_2}} (\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$

We try to show that if we already have a proof for $k$-equipartition for $n$ and if we know that $(n+1,\ldots,n+k)$ is a partition, then $k$-equipartition holds for $n+k$.

The end cedent of $P_n$ is

$$\bigwedge_{1\leqslant i\leqslant n}\bigwedge_{\substack{\vec{j}\in J^k_n\\ i\in\vec{j}}} x_{i,\vec{j}},\ \bigwedge_{\vec{j}\in J^k_n, i_1,i_2\in\vec{j}, i_1\neq i_2} (\bar{x}_{i_1,\vec{j}} x_{i_2,\vec{j}}),\ \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_n\\ i\in\vec{j}_1, i\in\vec{j}_2, \vec{j}_s 1\neq\vec{j}_2}} (\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$$

Use logical inference of which right upper cedents are

$$x_{i,\vec{j}}, \bar{x}_{i,\vec{j}}$$

for all $1\leqslant i\leqslant n$, $\vec{j}\in J^k_{n+k}-J^k_n$ and $i\in\vec{j}$. Their auxiliary literals are $x_{i,\vec{j}}$. Then we obtain the cedent;

$$\bigwedge_{\substack{1\leqslant i\leqslant n\\ \vec{j}\in J^k_{n+k}-J^k_n, i\in\vec{j}}} \bar{x}_{i,\vec{j}},\ \bigwedge_{1\leqslant i\leqslant n}\bigvee_{\vec{j}\in J^k_{n+k}, i\in\vec{j}} x_{i,\vec{j}},\ \bigwedge_{\vec{j}\in J^k_n, i_1,i_2\in\vec{j}, i_1\neq i_2} (\bar{x}_{i_1,\vec{j}} x_{i_2,\vec{j}}),\ \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_n\\ i\in\vec{j}_1, i\in\vec{j}_2, \vec{j}_1\neq\vec{j}_2}} (\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$$

Use structural inference to obtain the cedent

$$\bigwedge_{\substack{1\leqslant i\leqslant n\\ \vec{j}\in J^k_{n+k}-J^k_n, i\in\vec{j}}} \bar{x}_{i,\vec{j}},\ \bigwedge_{1\leqslant i\leqslant n+k}\bigvee_{\vec{j}\in J^k_{n+k}, i\in\vec{j}} x_{i,\vec{j}},\ \bigwedge_{\vec{j}\in J^k_{n+k}, i_1,i_2\in\vec{j}, i_1\neq i_2} (\bar{x}_{i_1,\vec{j}} x_{i_2,\vec{j}}),\ \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_{n+k}\\ i\in\vec{j}_1, i\in\vec{j}_2, \vec{j}_1\neq\vec{j}_2}} (\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$$

Note that for $x_{i,\vec{j}}$ such that $\vec{j}\in J^k_{n+k}-J^k_n$, there exists at least one $m$ satisfying that $n+1\leqslant m\leqslant n+k$ and $m\in\vec{j}$. If $i$ is included in a partition $\vec{j}$, then $m$ is also included in it. Define $J^*$ by the set of vectors on $\{1,\ldots,n,n+k\}$ and $\vec{t}=(n+1,\ldots,n+k)$. Use logical inferences of which right upper cedents are

$$x_{i,\vec{j}},\ \bar{x}_{i,\vec{j}}$$

for all $n+1\leqslant i<n+k, \vec{j}\in J^k_{n+k}-\{\vec{t}\}$ and $i\in\vec{j}$. Their auxiliary literals are $x_{i,\vec{j}}$.

$$\bigwedge_{\substack{n+1\leqslant i<n+k\\ \vec{j}\in J^k_{n+k}-\{\vec{t}\}, i\in\vec{j}}} \bar{x}_{i,\vec{j}},\ \bigwedge_{1\leqslant i\leqslant n, \vec{j}\in J^*} \bar{x}_{i,\vec{j}},$$

$$\bigwedge_{1\leqslant i\leqslant n+k}\bigvee_{\vec{j}\in J^k_{n+k}, i\in\vec{j}} x_{i,\vec{j}},\ \bigwedge_{\vec{j}\in J^k_{n+k}, i_1,i_2\in\vec{j}, i_1\neq i_2} (\bar{x}_{i_1,\vec{j}} x_{i_2,\vec{j}}),\ \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_{n+k}\\ i\in\vec{j}_1, i\in\vec{j}_2, \vec{j}_1\neq\vec{j}_2}} (\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$$

If $\vec{t}$ is a partition, then for all $i(n+1\leqslant i<n+k)$, $i$ is not included in any partition $\vec{j}$ such that $\vec{j}\neq\vec{t}$. Use logical inferences of which right upper cedents are

$$x_{i,\vec{t}}, \bar{x}_{i,\vec{t}},$$

where $n+1 \leqslant i < n+k$ and its auxiliary literal is $\bar{x}_{i,\vec{t}}$. Then we obtain the cedent;

$$\bigwedge_{n+1\leqslant i<n+k} x_{i,\vec{t}},\ \bigwedge_{\substack{n+1\leqslant i<n+k\\ \vec{j}\in J^k_{n+k}-\{\vec{t}\},i\in\vec{j}}} x_{i,\vec{j}}\bar{x}_{i,\vec{t}}\ \bigwedge_{1\leqslant i\leqslant n,\vec{j}\in J^*} \bar{x}_{i,\vec{j}},$$

$$\bigwedge_{1\leqslant i\leqslant n+k}\bigvee_{\vec{j}\in J^k_{n+k},i\in\vec{j}} x_{i,\vec{j}},\ \bigwedge_{\vec{j}\in J^k_{n+k},i_1,i_2\in\vec{j},i_1\neq i_2}(\bar{x}_{i_1,\vec{j}}x_{i_2,\vec{j}}),\ \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_{n+k}\\ i\in\vec{j}_1,i\in\vec{j}_2,\vec{j}_1\neq\vec{j}_2}}(\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$$

Use a structural inference to obtain the following cedent;

$$\bigwedge_{n+1\leqslant i<n+k} x_{i,\vec{t}},\ \bigwedge_{1\leqslant i\leqslant n,\vec{j}\in J^*}\bar{x}_{i,\vec{j}},$$

$$\bigwedge_{1\leqslant i\leqslant n+k}\bigvee_{\vec{j}\in J^k_{n+k},i\in\vec{j}} x_{i,\vec{j}},\ \bigwedge_{\vec{j}\in J^k_{n+k},i_1,i_2\in\vec{j},i_1\neq i_2}(\bar{x}_{i_1,\vec{j}}x_{i_2,\vec{j}}),\ \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_{n+k}\\ i\in\vec{j}_1,i\in\vec{j}_2,\vec{j}_1\neq\vec{j}_2}}(\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$$

Use logical inferences of which right upper cedents are

$$x_{n+k,\vec{j}},\bar{x}_{n+k,\vec{j}},$$

where $\vec{j}\in J^*$, and its auxiliary literal is $x_{n+k,\vec{j}}$. Then we obtain the cedent

$$\bigwedge_{\vec{j}\in J^*,n+k\in\vec{j}}\bar{x}_{n+k,\vec{j}},\ \bigwedge_{n+1\leqslant i<n+k} x_{i,\vec{t}},$$

$$\bigwedge_{1\leqslant i\leqslant n+k}\bigvee_{\vec{j}\in J^k_{n+k},i\in\vec{j}} x_{i,\vec{j}},\ \bigwedge_{\vec{j}\in J^k_{n+k},i_1,i_2\in\vec{j},i_1\neq i_2}(\bar{x}_{i_1,\vec{j}}x_{i_2,\vec{j}}),\ \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_{n+k}\\ i\in\vec{j}_1,i\in\vec{j}_2,\vec{j}_1\neq\vec{j}_2}}(\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$$

If $\vec{t}$ is a partition, then $n+k$ is not included in any partition $\vec{j}$ such that $\vec{j}\neq\vec{t}$. Use logical inference of which right upper cedent is

$$x_{n+k,\vec{t}},\bar{x}_{n+k,\vec{t}},$$

where its auxiliary literal is $\bar{x}_{n+k,\vec{t}}$. Then we obtain the cedent

$$x_{n+k,\vec{t}},\ \bigwedge_{1\leqslant i\leqslant n+k}\bigvee_{\vec{j}\in J^k_{n+k},i\in\vec{j}} x_{i,\vec{j}},\ \bigwedge_{\substack{\vec{j}\in J^k_{n+k},i_1,i_2\in\vec{j}\\ i_1\neq i_2}}(\bar{x}_{i_1,\vec{j}}x_{i_2,\vec{j}}),\ \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_{n+k}\\ i\in\vec{j}_1,i\in\vec{j}_2,\vec{j}_1\neq\vec{j}_2}}(\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$$

This cedent means that if $k$-equipartition holds for $n$, and, furthermore, if we already know that $(n+1,n+2,\ldots,n+k)$ is a partition of $\{1,\ldots,n,n+1,\ldots,n+k\}$, then $k$-equipartition holds for $n+k$. Denote this cedent by $C$. For every $\vec{j}=(j_1,\ldots,j_k)\in J^k_{n+k}-\{\vec{t}\}$ such that $n+k\in\vec{j}$, define a permutation $\pi_{\vec{j}}$ by a transposition,

$$(x_{n+k,\vec{j}}\quad x_{n+k,\vec{t}}).$$

Apply each permutation $\pi_{\vec{j}}$ on $C$ to obtain the cedents;

$$x_{n+k,\vec{j}},\ \bigwedge_{1\leqslant i\leqslant n+k}\bigvee_{\vec{j}\in J^k_{n+k},i\in\vec{j}} x_{i,\vec{j}},\ \bigwedge_{\vec{j}\in J^k_{n+k},i_1,i_2\in\vec{j}\,i_1\neq i_2}(\bar{x}_{i_1,\vec{j}}x_{i_2,\vec{j}}),\ \bigwedge_{\substack{\vec{j}_1,\vec{j}_2\in J^k_{n+k},\\ i\in\vec{j}_1,i\in\vec{j}_2,\vec{j}_1\neq\vec{j}_2}}(\bar{x}_{i,\vec{j}_1}\bar{x}_{i,\vec{j}_2})$$

Finally, use logical inferences to combine them together to obtain a refutation of $P_{n+k}$ of $k$–$Eq(n+k)$. The size of $P_{n+k}$ is polynomially bounded. After close examination, we can also conclude that if $P_n$ is regular, then so is $P_{n+k}$.

**Corollary 1.** *Resolution does not p-simulate* GCNF′+*permutation.*

**Corollary 2.** *Bounded depth Frege systems do not p-simulate* GCNF′+ *permutation.*

## 3. The consistency of Frege + renaming

In this section, we discuss the relative efficiency of GCNF′+ renaming (or Frege + renaming) with Frege or extended Frege systems. There are two different motivations to scrutinize renaming rule. We showed that GCNF′ + renaming polynomially simulate GCNF′ + permutation. Hence, it is helpful to check the relative efficiency of GCNF′ + renaming especially with Frege (or equivalently with Gentzen system, LK) in finding the lower bound for GCNF′ + permutation. The second motivation comes from the separation problem of Frege system and extended Frege system. It is counted as one of the most important questions in the field of computational complexity whether or not P = ALOGTIME. Earlier works by Cook [10] and Buss [7] suggest that extended Frege systems correspond to P whereas Frege systems correspond to ALOGTIME. Later Buss showed that Frege+renaming $p$-simulates extended Frege [4]. Hence, it will be useful to analyze the characters of renaming rule to answer the separation problem of these two systems.

**Definition 6.** We define a system for propositional calculus, called *Frege systems*, denoted by $F$. $F$ consist of

1. A language $L$, a finite complete set of propositional connectives.
2. A finite set of axiom schemata.
3. A proof will be a sequence of propositions $A_1,\ldots,A_n$, where each $A_i$ is either a substitution instance of an axiom, or inferred by *modus ponens* from some $A_j$ and $A_k$, where $j,k \leqslant i$. Modus ponens is an inference rule which allows us to infer $\psi$ from $\varphi$ and $\varphi \supset \psi$.

$\{\neg, \wedge, \vee, \supset\}$ is an example of a complete set of propositional connectives.

**Definition 7.** *Extended Frege system* is a propositional calculus obtained by adding the following inference rule, called *extension rule* which allows introduction of abbreviations to Frege system: the extension rule allows the derivation of $p \leftrightarrow \varphi$ where $p$ is a new variable which has not been used yet in the proof and does not appear in $\varphi$ or in the final line of the proof.

Buss introduced a new inference rule called (0/1)-substitution, which allows renaming inference $(p \rightarrow \top)$ and $(p \rightarrow \bot)$. By demonstrating that Frege + (0/1)-substitution

$p$-simulates extended Frege system, and that Frege + renaming $p$-simulates Frege + (0/1)-substitution, he showed that Frege + renaming $p$-simulates extended Frege [4, 5].

**Lemma 1** (Buss [4]). *Frege + (0/1)-substitution p-simulates extended Frege.*

**Lemma 2** (Buss [4]). *Frege + renaming p-simulates Frege + (0/1)-substitution.*

**Proof.** Suppose that $A(x_1,\ldots,x_n)$ has a Frege + (0/1)-substitution proof $P$. A Frege + renaming proof $Q$ of $A(x_1,\ldots,x_n)$ can be constructed as follows. First, form Frege proofs $P_0$ of $A(0,\ldots,0)$ and $P_1$ of $A(1,\ldots,1)$. Second, form a Frege + (0/1)-substitution proof $P'$ of $A(y_1,\ldots,y_n)$, which does not use variables $x_1,\ldots,x_n$ at all. Let $Z$ be the statement

$$\neg(x_1 \wedge \cdots \wedge x_n) \wedge (x_1 \vee \cdots \vee x_n)$$

which says that $x_i$'s are not all true and not all false. Replace every line $B$ in $P'$ by $Z \supset B$. Now replace (0/1)-substitution inference in $P'$ as follows. If

$$\frac{Z \supset B(y)}{Z \supset B(0)}$$

is an inference in $P'$, use renaming rule $n$ times to derive

$$Z \supset B(x_i)$$

for all $1 \leqslant i \leqslant n$ from $Z \supset B(y)$. Combining them together by using propositional inferences, derive

$$Z \supset B(x_1) \wedge \cdots \wedge B(x_n).$$

From this $Z \supset B(0)$ is inferred, and so is $Z \supset B(1)$. In this way, a Frege + renaming proof of $Z \supset A(y_1,\ldots,y_n)$ can be obtained. Use renaming inferences $n$ times to get

$$Z \supset A(x_1,\ldots,x_n).$$

Using three proofs $P_0, P_1$ and $P'$, one can obtain a proof $Q$ of $A(x_1,\ldots,x_n)$.

Note that in proving this lemma it is essential to use renaming rule but not just restricted renaming rule or permutation rule.

**Theorem 5** (Buss [4]). *Frege + renaming p-simulates extended Frege.*

We follow the argument in [6] by Buss, and prove that Frege system has polynomial size proofs of partial consistency of Frege + renaming.

As shown in [6], Frege systems can perform metamathematics inside it. Let $\vec{x}$ represent a vector of propositional variables $x_1,\ldots,x_{ck}$, where $c$ is a constant so that we can code the symbols (including $p, 0, 1$, propositional connectives, parenthesis) by strings of $\top$'s and $\bot$'s. For example a propositional variable $p_i$ will be represented by the

code of $p$ (denoted by '$p$') followed by a string of 0's and 1's coding $i$ binary. Frege is able to formulate concepts such that 'formulas', 'proofs' and $Con_F(n)$ which means that there is no $F$-proof of $\bot$ (or equivalently $p_0 \wedge \neg p_0$) of size $n$, by polynomial size formulas. $\vec{x}[i]$ means the $i$th logical symbol in $\vec{x}$. $\vec{x}[i,j]$ denotes the substring of $\vec{x}$ from $\vec{x}[i]$ through $\vec{x}[j]$ inclusive.

**Lemma 3** (Buss [6]). *Let $\varphi$ be a formula in the language of $F$. Then,*

$$F \vdash^* \text{“}\vec{x}[i,j] \text{ encodes } \varphi \supset (TRUE(\vec{x}[i,j]) \leftrightarrow \varphi)\text{”}$$

*where $F \vdash^* \psi$ means that there is a proof whose size $\leqslant p(\text{size of } \psi)$ for some polynomial $p$. $TRUE(\vec{x}[i,j])$ denotes the polynomial size formula with variables $p_1, p_2, \ldots$ which may be named in the formula coded by $\vec{x}$ so that $TRUE(\vec{x}[i,j])$ is true iff the formula coded by $\vec{x}[i,j]$ is true.*

Note that $TRUE$ only mentions truth under some 'fixed assignment', but does not mention validity. Buss showed that Frege systems can prove their own partial consistency [6]. We extend his argument and prove that Frege systems can prove the partial consistency of Frege + renaming.

**Theorem 6.** *For every fixed $\varphi$,*

$$F \vdash^* \text{“}x \text{ is not a } F + \text{renaming proof of } \varphi \wedge \neg\varphi\text{.”}$$

**Proof.** We argue informally in $F$. Suppose that $\vec{x}$ is a $F+$ renaming proof of $\bot$. Replace every propositional variable appearing in $\vec{x}$ by $\bot$. Then, by brute force 'induction', we can prove that the obtained sequence, say $\vec{y}$ is a Frege proof of $\bot$. Then, again by brute force 'induction', we can show that any formula, say $\vec{x}[i,j]$ appearing in the proof coded by $\vec{y}$ is true i.e. $TRUE(\vec{y}[i,j])$. But it yields a contradiction since the last line of $\vec{y}$ is not true.

Strangely enough, $F$ $p$-simulates extended-$F$ iff $F$ proves the partial consistency of $F+(0/1)$-substitution. (In his original proof, Buss showed that $F$ $p$-simulates extended-$F$ iff $F$ proves the partial consistency of extended-$F$ [6]). Where does the difference lie, proving the partial consistency of $F+$ renaming and that of $F+(0/1)$-substitution?

**Theorem 7.** *Frege system p-simulates extended Frege system if and only if*

$$F \vdash^* \text{“}\vec{x} \text{ is not a } F + (0/1)\text{-substitution proof of } \bot\text{.”}$$

We try to demonstrate the difference between proving the partial consistency of $F+$ renaming and that of $F+(0/1)$-substitution by the following corollary.

**Corollary 3.** *Frege proof system p-simulates extended Frege system if and only if*

$$F \vdash^* \text{“}\vec{x} \text{ is not a } F + \text{renaming proof of } p \supset q\text{.”}$$

**Proof.** It suffices to show that $F \vdash^*$ "$\vec{x}$ *is not a* $F+$*renaming proof of* $p \supset q$" is equivalent to $F \vdash^*$ "$\vec{x}$ *is not a* $F + (0/1)$-*substitution proof* of $p \supset q$". We argue informally within the polynomial size proof of Frege system. Let $P$ be a $F + (0/1)$-substitution proof of $\bot$. Replace each line $B$ in $P$ by $B'$, where $B'$ is $B$ except that every occurrence of $\bot$ is replaced by $q$ and that of any variable or 1 is replaced by $p$. It may fail to be a valid $F+$renaming proof. Replace every line $B'$ by $p \supset (B' \vee q)$. Then we obtain a $F+$renaming proof of $p \supset (q \wedge q)$, which is equivalent to $p \supset q$. Note that the procedure can be carried out in $F$. The converse is proved likewise.

The difficulty of disproving the existence of $F+$renaming proof of $p \supset q$ lies in the fact that we have to mention validity but not just truth under a specified assignment when we try to show that $p \supset q$ is not valid. Frege system only enables us to deal with TRUTH of boolean formulas under a given assignment. With a help of renaming rule, we are able to mention validity. Is renaming rule solely that powerful, or is it only powerful when cut rule is available?

**Conjecture.** GCNF′+renaming does not $p$-simulate Frege system.

## Acknowledgements

I would like to express my sincere gratitude to Samuel Buss for his encouraging comments and useful advices. I am also grateful to anonymous referees for their helpful suggestions which improved this paper.

## References

[1] M. Ajtai, The complexity of the pigeonhole principle, *29th Annual Symp. On the Foundations of Computer Science* (1988) 346–55.
[2] B. Benhamou and L. Sais, Tractability through symmetries in propositional calculus, *J. Autom. Reasoning* **12** (1994) 89–102.
[3] S.R. Buss, Polynomial size proofs of the pigeonhole principle, *J. Symbol. Logic* **52** (1987) 916–927.
[4] S.R. Buss, private communication, 1993.
[5] S.R. Buss et al., *Weak Formal Systems and Connections to Computational Complexity*, Student-written lecture notes for topics course at U.C. Berkley, January–May, 1988.
[6] S.R. Buss, Propositional consistency proofs, *Ann. Pure Appl. Logic* **52** (1991) 3–29.
[7] S.R. Buss, *Bounded Arithmetic* (Bibliopolis, Napoli, 1986).
[8] P. Clote, On polynomial size Frege proofs of certain combinatorial principles, in: *Arithmetic, Proof Theory, and Computational Complexity* (Clarendon Press, Oxford, 1993) 162–184.
[9] S.A. Cook and R.A. Reckhow, The relative efficiency of propositional proof systems, *J. Symbol. Logic* **44** (1979) 36–50.
[10] S.A. Cook, Feasibly constructive proofs and the propositional calculus, *Proc. 7th A.C.M. Symp. on the Theory of Computation* (1975) 83–97.
[11] J. Gallier, *Logic for Computer Science* (Wiley, New York, 1987).
[12] A. Haken, The intractability of resolution, *Theoret. Comput. Sci.* **39** (1985) 297–308.

[13] G.S. Tseitin, On the complexity of derivation in propositional calculus, *Studies in Mathematics and Mathematical Logic Part 2*, V.A. Steklov Math. Institute, 1968.
[14] A. Urquhart, The complexity of Gentzen systems for propositional logic, *Theoret. Comput. Sci.* **66** (1989) 87–97.
[15] A. Urquhart, Hard examples for resolution, *J. Assoc. Comput. Mach.* **34** (1987) 209–219.